

We claim:

1. A programmable processor comprising:

a data path;

an external interface operable to receive data from an external source and communicate
5 the received data over the data path;

a cache operable to retain data communicated between the external interface and the data
path;

a register file coupled to the data path and containing a plurality of registers; and

an execution unit coupled to the data path, the execution unit configurable to perform a
10 group instruction that operates on a plurality of data elements in partitioned fields of a
register to produce a catenated result, the execution unit further configurable to execute:

(i) an aligned instruction operable to copy first data according to an aligned memory
address, the first data having a data width, the data width specified as a fixed value by the
aligned instruction, the aligned memory address being one of a plurality of memory
15 addresses regularly spaced at alignment boundaries separated by the data width; and

(ii) an unaligned instruction operable to copy second data according to an unaligned
memory address, the second data having the data width, the data width specified as a fixed
value by the unaligned instruction, the second data being permitted to cross an alignment
boundary of the data width, the unaligned memory address being a memory address that is
20 not constrained to be one of the plurality of memory addresses regularly spaced at alignment
boundaries separated by the data width.

2. The processor of claim 1 wherein the aligned instruction comprises a load instruction
operable to copy the first data from memory at the aligned memory address to a register, and
the unaligned instruction comprises a load instruction operable to copy the second data from
25 memory at the unaligned memory address to a register.

3. The processor of claim 1 wherein the aligned instruction comprises a store instruction
operable to copy the first data from a register to memory at the aligned memory address, and
the unaligned instruction comprises a store instruction operable to copy the second data from
a register to memory at the unaligned memory address.

4. The processor of claim 1 wherein the group instruction is capable of operating on data having a data width of 128 bits.
5. The processor of claim 1 wherein the group instruction is a group floating-point instruction.
6. The processor of claim 1 wherein the group instruction is a group integer instruction.
- 5 7. The processor of claim 1 wherein the aligned instruction and the unaligned instruction are capable of accessing the first data and the second data, each having the data width of 128 bits
8. The processor of claim 1 wherein the aligned instruction and the unaligned instruction are capable of accessing the first data and the second data, each having the data width of 64 bits.
9. The processor of claim 1 wherein the plurality of regularly spaced memory addresses are
10 separated by intervals of 128 bits.
10. The processor of claim 1 wherein the plurality of regularly spaced memory addresses are separated by intervals of 64 bits.
11. The processor of claim 1 wherein the aligned instruction responds by generating an exception if the aligned memory address is not one of a plurality of memory addresses regularly spaced
15 at alignment boundaries separated by the data width.
12. The processor of claim 1 wherein the execution unit is further configurable to execute two aligned instructions in parallel using hardware capable of executing a single unaligned instruction.
13. The processor of claim 1 wherein the aligned instruction corresponds to a first binary code
20 and the unaligned instruction corresponds to a second binary code, the first binary code matching the second binary code in all but one bit position.

14. A method of providing data and memory capabilities in a programmable processor, the method comprising:

providing, in an instruction set for the processor, a group instruction that operates on a plurality of data elements in partitioned fields of at least one register to produce a catenated result;

providing, in the instruction set for the processor, an aligned instruction operable to copy first data according to an aligned memory address, the first data having a data width, the data width specified as a fixed value by the aligned instruction, the aligned memory address being one of a plurality of memory addresses regularly spaced by the data width; and

providing, in the instruction set for the processor, an unaligned instruction operable to copy second data according to an unaligned memory address, the second data having the data width, the data width specified as a fixed value by the unaligned instruction, the unaligned memory address being a memory address that is not constrained to be one of the plurality of memory addresses regularly spaced by the data width.

15. The method of claim 14 wherein the aligned instruction comprises a load instruction operable to copy the first data from memory at the aligned memory address to a register, and the unaligned instruction comprises a load instruction operable to copy the second data from memory at the unaligned memory address to a register.

16. The method of claim 14 wherein the aligned instruction comprises a store instruction operable to copy the first data from a register to memory at the aligned memory address, and the unaligned instruction comprises a store instruction operable to copy the second data from a register to memory at the unaligned memory address.

17. The method of claim 14 wherein the group instruction is capable of operating on data having a data width of 128 bits.

18. The method of claim 14 wherein the group instruction is a group floating-point instruction.

19. The method of claim 14 wherein the group instruction is a group integer instruction.

20. The method of claim 14 wherein the aligned instruction and the unaligned instruction are capable of accessing the first data and the second data, each having the data width of 128 bits

21. The method of claim 14 wherein the aligned instruction and the unaligned instruction are capable of accessing the first data and the second data, each having the data width of 64 bits.
22. The method of claim 14 wherein the plurality of regularly spaced memory addresses are separated by intervals of 128 bits.
- 5 23. The method of claim 14 wherein the plurality of regularly spaced memory addresses are separated by intervals of 64 bits.
24. The method of claim 14 wherein the aligned instruction responds by generating an exception if the aligned memory address is not one of a plurality of memory addresses regularly spaced at alignment boundaries separated by the data width.
- 10 25. The method of claim 14 wherein two aligned instructions are capable of parallel execution.
26. The method of claim 14 wherein the aligned instruction corresponds to a first binary code and the unaligned instruction corresponds to a second binary code, the first binary code matching the second binary code in all but one bit position.